TOWARDS IMPROVING SOFTWARE DEVELOPERS PRODUCTIVITY AND EFFECTIVENESS

[1]S.C. Chiemeke & [2]F.A. Egbokhare
Department of Computer Science,
University of Benin
Benin City, Nigeria
[1]schiemeke@uniben.edu, fegbokhare@yahoo.com

[1]Corresponding Author

ABSTRACT

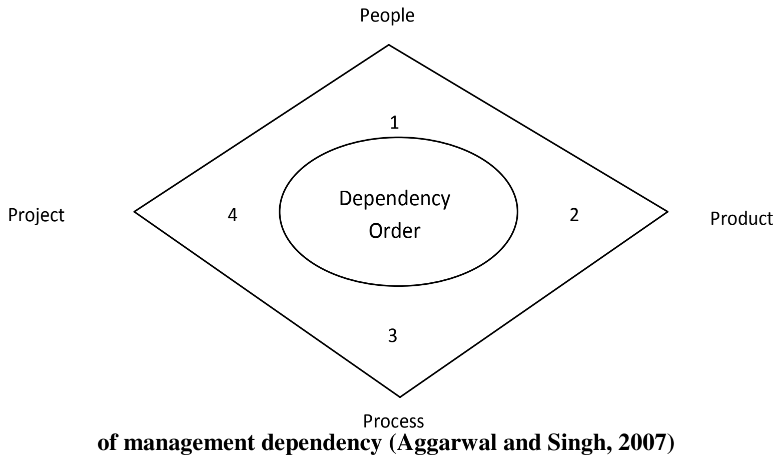
Computer software has become an engine that drives business decision making process. Studies on the challenges facing software development organizations in Nigeria revealed certain human-centric and environmental factors, coupled with unwillingness by most software project managers to invest in recent technologies. These resulted in hostilities and chaos in such organizations, translating to low productivity and unusable software products. In this paper, we critically examined some actions taken at the University of Benin to make software developers more productive, effective and satisfied.

Keywords:

1. INTRODUCTION

Computer software has become an engine that drives business decision making hence huge amounts of funds are invested into Software development projects. Most of these projects however especially in developing countries suffer chronically from cost overruns, project delays, unmet user needs and unusable/abandoned systems. This has continued despite huge advances in development techniques, tools and software technologies (Griffith and Newman, 1996). In the early 90s, most failure factors were traced to problems in the software development process and various frameworks were developed to improve the software development process (Humphrey, 1990; Paulk et al, 1993). A process is the way in which we produce software. It is the collection of activities (actions), resources (human and material) and constraints (cost, time) that leads to a product (software system). Paulk et. al. (1993) defined software process as a set of activities, methods, practices and transformations that people use to develop and maintain software and the associated products. It encompasses the total set of engineering activities needed to transform a user’s requirement into software system (Humphrey, 1990).

A software process therefore leads to design, development and implementation of a software system. The quality of software system depends on the quality of the processes used to produce and maintain it, hence if the process is weak, the end product will undoubtedly suffer, but an obsessive over reliance on process is also dangerous (Humphrey, 1990). Aggarwal and Singh (2007) identified four major factors that influence the management of software development: People, Product, Process and Project. The order of dependency of these factors is as shown in Figure 1.



of management dependency (Aggarwal and Singh, 2007)

From the figure 1, we can deduce that software development is people-centric and this is consistent with Passova (2005), who viewed the software development process as a multi-level human communication channel where the problem originates from people (customers), creative and knowledgeable people (developers) use available tools and technology (process) to build the required software systems and the process is also managed by people (manager). Egbokhare and Onibere (2008) noted that most software engineering literature focus primarily on the process and the project and identified neglect of the people factor in software development projects as one of the major causes of failures in software development projects. Because the human resource is volatile and unpredictable, failure to direct it at the right time and cost often leads to serious inefficiencies such as low productivity and serious catastrophes especially in large and complex projects such as software development (Weinberg, 2004; Royce, 2005, Akinyokun and Chiemeke, 2006; Beardwell and Holder, 1997).

The unsettling revelation in published analysis of software development show that the percentage of projects that fail by coming in far over budget or far past schedule or by being cancelled with significant financial loss is considerably greater in software development than any part of engineering. Chiemeke (2000) suggested increase in developers' domain knowledge and adaptive change management as factors that can improve developers productivity and reduce software project failures. Also, Egbokhare and Onibere(2010) proposed a framework for human-level interactions that can constantly aid software project managers in the identification of both process and non-process related factors that can be consistently reviewed to reduce software project failures especially in developing countries.

2. SOFTWARE DEVELOPMENT AND UTILIZATION AT THE UNIVERSITY OF BENIN

The University of Benin is one of the forerunners of Information and Communications Technology (ICT) implementation and deployment in educational establishments in Nigeria. To effectively manage the Institution's information system, the Central Record processing Unit (CRPU) was established and staffed with software developers (Programmers, Systems Analysts, Web Designers and Database designers) under the office of the Institution's Director of ICT.

To overcome some of the problems relating to students course registration and all forms of fees payment, Management (in an attempt to fulfill one of her main objectives to keep students abreast with the latest technology) introduced the Student Registration Portal in the 2003/2004 academic session. Despite the fact that there is a team of qualified software developers in the institution, external contractors had been hired over the years to design, implement and maintain the University portals.
Since these contractors do not interact directly with the end users to gather the actual requirements for the system, users of the system are unsatisfied and today, the portal users (Mostly students and course level advisers) have raised several concerns and issues which cannot be directly handled by the internal software development team because they lack access to the source code and documentation for the portal. The current ICT Director, thus in an attempt to salvage the situation sought ways to improve developers productivity while attempting to tackle some of the issues raised by users.

3. METHODOLOGY

A single case study was used in this paper and the case was the Institution's Portal and the extent to which the CRPU section, the department responsible for the management of the application programs, management of student records and the databases had been able to effectively these activities. Since the developers (comprising systems analysts, database designers, programmers and testers) are directly incharge of the portal services, they were used as the focus group for this study. The human resource interaction model proposed in Egbokhare and Onibere (2010) served as a framework for Informal interactions with the Institution's software developers.

This model actually defines framework that can be used as a tool to gather data from developers during software development projects. The main purpose of the interaction with the developers was to brainstorm on issues leading to the current low level of productivity by software developers and how to resolve in-house some of the issues raised by the users of the system. Factors that contributed to the failure of most of the projects were identified as:

- Total isolation of developers from the institutions' software development project isolation from
- Lack of confidence and trust on the ability of developers
- Lack of motivation especially in the area of manpower development and availability of tools to do the job
- Bureaucratic bottlenecks that indirectly saddled developers with more administrative and paper work rather than code production and management
- Lack of interaction between system developers and users leading to difficult to use systems
- Lack of source code and system documentation
- Lack of continuity in policy
- Lack of Institutional support

4. DISCUSSION

A cursory look at these factors indicate that the primary cause of low productivity of software developers can be blamed on lack of motivation and institutional support. In this paper, we define productivity as the rate at which quality software that meet user requirements can be developed. We discovered that the software developers are isolated from the software development process. This leads to the development of unuseable systems, since the first contact between developers and university portal is at the testing stage (Two University Portals were abandoned due to improper definitions of requirements at conceptual the stage). Also because the managers are more interested in the financial aspect of the project thus they sign contracts oblivious of the intricacies of designs and code generations. This dampened the enthusiasm of the internal software developers which resulted in ineffectiveness, lack of confidence and trust in the Institution's Management culminating in their inability to perform optimally.
Since July 2010 when the current ICT Director (who is also one of the authors of this paper) took over the headship of the Institution's ICT centre, several programmes were initiated to resolve some of the findings above especially those relating to developers' motivation and improving their productivity. Two major reforms were embarked on.

The first was to upgrade the developers' knowledge through local and international conferences, workshops organized specifically for software developers and seminars. This was to rejuvenate their zeal to become system developers and code managers again so that thy can handle the Institution's software development projects internally to save cost, minimize downtime, and make the Institution's operations more effective.

Secondly, we regrouped the developers and formed them into sub-teams to attempt to develop a new portal for the Institution. Since this group had been data custodians and data managers to the Institution for about ten (10) years, they understand the nature of inputs and outputs required in the Institution. Also, because the CRPU staff had interacted so much with the system users, there was a huge documentation on what the users expect from the portal and the complex services that need to be readdressed. The documented requirements over the years as well as fresh requirements from the various application domains were used to prepare the specification for the system. Roles and responsibilities were defined and assigned. Timelines were set and the sub-teams met every fortnight for periodic reviews and to discuss project related issues. This project has been on for about nine (9) months and today, the portal is at the testing phase.

5. CONCLUSION

The main objective of this paper was to motivate software developers and challenge them to become more productive and effective. The actions taken so far helped to boost the staff morale, uplift their confidence level and general sense of belonging. Today, we can boast that the level of productivity of the Institution's developers has risen well above zero and that the developers are prepared to take up more challenging jobs. The in-house portal being developed has not yet been presented to management but we believe it will be accepted and since the source code and documentation is available, maintaining the system will be easier.

REFERENCES

1. Aggarwal, K. K. and Singh, Y. (2007). Software Engineering. New Age International Publishers, New Delhi. 480pp
2. Akinyokun, O.C. and Chiemeke S.C. (2006). IT Projects Performance Indices and Evaluation. *Journal of Testing and Evaluation*. 34(6). ISSN: 0090-3973.
3. Chiemeke, S. C. (2000). "Current Issues in Software Engineering" *Journal of Science and Mathematics in Southeast Asia* Vol. XII No. 45 pp. 1193 – 1200.
4. Egbokhare F.A.and Onibere, E.A. (2008) "The Most Neglected 'P' in Software Development Projects. *Asian Journal of Information Technology* Vol.7, No. 9pp 416-419. (Pakistan)
5. Egbokhare F.A. and Onibere, E.A. (2010) "Intrinsic Software Management". *Journal of Institute of Mathematics and Computer Sciences (Computer Science Series*). 121(2), 237 – 244 India.
6. Beardwell, I and Holder, L. (1997). Human Resource Management: A Comtemporary Perspective. Pitman Publishing, England. 816pp
7. Humphrey, W.S.(1990), Managing The Software Process. Addison –Wesley. 494pp
8. Onibere, E.A. and Egbokhare, F.A.(2007). Software Development Process: A Case of Nigerian Software Development Organizations. *The Information Technologist. Vol.4 No. 1 pp. 1-13*
9. Passova, S.(2005). The Need for Customer-Oriented Software Development. Sofea, Inc. e-Biz Development tools. Available online at: www.ebiz.net/topics/devtools/features/5577.html
10. Paulk M, Weber C, Garcia S., Chrissis M. and Bush M.(1993a). Capability Maturity Model For Software, version 1.1. SEI-93-TR-024. Software Engineering Institute
11. Royce, W.(2005). Successful Software Management Style: Steering and Balance. IEEE Computer Society, 40-47.
12. Weinberg, G.M. (2004). Advice for Software Managers. Available Online at WWW.geraldmweinberg.com.

Authors' Briefs

Prof. (Mrs.) Stella Chinye CHIEMEKE is a Professor of Computer Science and currently the Director of UNIBEN ICT Centre, University of Benin, Benin City, Nigeria. Her teaching and research interests spans from software engineering to industrial application of ICT. She is a member of the International Association of Engineers (IAENG), Computer Professional of Nigeria (CPN,)Nigeria Computer Society (NCS), International Network for Woman Engineers and Scientists (INWES) Canada etc.

Dr(Mrs) F.A Egbokhare holds a PhD in Computer Science. She is a Senior Lecturer in the Department of Computer Science, University of Benin. Current research area is Human Resource management in Software Projects. Dr Egbokhare is currently the Departmental PhD coordinator.